

jQuery

Day01

课上练习

暂无

案例

1.5 案例 - JQ版 - 点谁谁亮

一排按钮-点击谁谁高亮

之前webAPI有个tab栏切换 webAPI/Day02 - 改成jq实现

```
$("#myWrap>input").on("click", function(){
    $(this).css({backgroundColor: "red"}).siblings().css({backgroundColor: ""})
})
```

1.7 案例 - JQ版 - tab栏

之前webAPI有个tab栏切换 webAPI/Day02 - 改成jq实现

```
// 1. 给每个li, 绑定点击事件
$(".tab_list li").click(function () {
    // 2. 当前自己加类名, 兄弟们移出类名
    $(this).addClass("current")
    $(this).siblings().removeClass("current");

    // 3. 一一对应! JQ对象.index() 返回目标元素下标
    var index = $(this).index();
    // 4. 用下角标取出对应的div显示, 其余的隐藏
    // $(".item").eq(index).css("display", "block");
    // $(".item").eq(index).siblings().css("display", "none");

    // 或者这样写更好 - 相当于简化了display:none和block写法
    $(".item").eq(index).show().siblings().hide();
});
```

2.1 案例 - JQ版 - 移入移出导航



微博

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>案例_JQ版_移入移出导航</title>
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        li {
            list-style-type: none;
        }

        a {
            text-decoration: none;
            font-size: 14px;
        }
    </style>

```

```

.nav {
    margin: 100px;
}

.nav>li {
    position: relative;
    float: left;
    width: 80px;
    height: 41px;
    text-align: center;
}

.nav li a {
    display: block;
    width: 100%;
    height: 100%;
    line-height: 41px;
    color: #333;
}

.nav>li>a:hover {
    background-color: #eee;
}

.nav ul {
    display: none;
    position: absolute;
    top: 41px;
    left: 0;
    width: 100%;
    border-left: 1px solid #FECC5B;
    border-right: 1px solid #FECC5B;
}

.nav ul li {
    border-bottom: 1px solid #FECC5B;
}

.nav ul li a:hover {
    background-color: #FFF5DA;
}
</style>

</head>

<body>
    <ul class="nav">
        <li>
            <a href="#">微博</a>
            <ul>
                <li>
                    <a href="">私信</a>

```

```
        </li>
        <li>
            <a href="">评论</a>
        </li>
        <li>
            <a href="">@我</a>
        </li>
    </ul>
</li>
<li>
    <a href="#">微博</a>
    <ul>
        <li>
            <a href="">私信</a>
        </li>
        <li>
            <a href="">评论</a>
        </li>
        <li>
            <a href="">@我</a>
        </li>
    </ul>
</li>
<li>
    <a href="#">微博</a>
    <ul>
        <li>
            <a href="">私信</a>
        </li>
        <li>
            <a href="">评论</a>
        </li>
        <li>
            <a href="">@我</a>
        </li>
    </ul>
</li>
<li>
    <a href="#">微博</a>
    <ul>
        <li>
            <a href="">私信</a>
        </li>
        <li>
            <a href="">评论</a>
        </li>
        <li>
            <a href="">@我</a>
        </li>
    </ul>
</li>
</ul>
```

```
</body>
<script src="../../js/jquery3.5.1.min.js"></script>
<script>

</script>

</html>
```

正确js代码

```
// 需求：鼠标移入哪个li,当前li 下面ul进行显示
//      鼠标移出哪个li, 当前li 下面ul进行隐藏

// 思路转变：只是针对当前li进行操作;
$(".nav>li").mouseover(function() {
    $(this).children("ul").show(200);
});
$(".nav>li").mouseout(function() {
    $(this).children("ul").hide(200);
});
```

2.3 案例 - JQ版 - 滑出导航

效果演示:



基于上个案例的html和css, 新的代码如下:

```
// 只是切换的效果, 使用不同的方法即可
$(".nav>li").mouseover(function () {
    $(this).children("ul").slideDown(300);
});
$(".nav>li").mouseout(function () {
    $(this).children("ul").slideUp(300);
});
```

2.7 案例 - JQ版 - 淡入淡出效果图

素材在images/看序号



标签和样式模板:

```
<!DOCTYPE html>
<html>

<head lang="en">
  <meta charset="UTF-8">
  <title></title>
  <style type="text/css">
    * {
      margin: 0;
      padding: 0;
    }

    ul {
      list-style: none;
    }

    body {
      background: #000;
    }

    .wrap {
      margin: 100px auto 0;
      width: 630px;
      height: 394px;
      padding: 10px 0 0 10px;
      background: #000;
```

```

        overflow: hidden;
        border: 1px solid #fff;
    }

    .wrap li {
        float: left;
        margin: 0 10px 10px 0;
    }

    .wrap img {
        display: block;
        border: 0;
    }
</style>

</head>

<body>
    <div class="wrap">
        <ul>
            <li>
                <a href="#"></a>
            </li>
            <li>
                <a href="#"></a>
            </li>
            <li>
                <a href="#"></a>
            </li>
            <li>
                <a href="#"></a>
            </li>
            <li>
                <a href="#"></a>
            </li>
            <li>
                <a href="#"></a>
            </li>
        </ul>
    </div>
    <script src="./js/jquery3.5.1.min.js"></script>
</body>

</html>

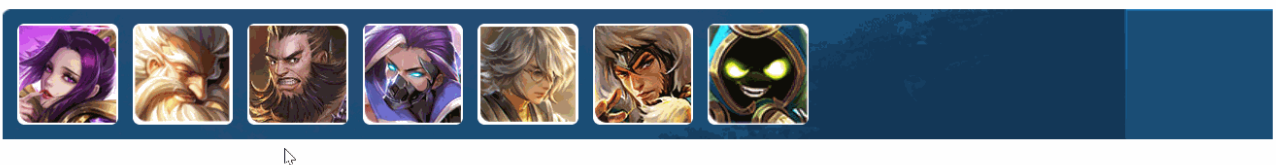
```

正确js功能代码:


```
$(function() {
    // 移入: 兄弟们变暗
    $("li").mouseenter(function() {
        $(this).siblings().stop().fadeTo(200, 0.3);
    });
    $("li").mouseleave(function() {
        $(this).siblings().stop().fadeTo(200, 1);
    });
});
```

3.1 案例 - JQ版 - 查看英雄介绍

素材在images/看序号



标签和样式模板:

```
<!doctype html>
<html>

<head>
    <meta charset="utf-8">
    <title>案例_JQ版_英雄的介绍</title>

    <style type="text/css">
        * {
            margin: 0;
            padding: 0;
        }

        img {
            display: block;
        }

        ul {
            list-style: none;
        }

        .king {
            width: 852px;
            margin: 100px auto;
            background: url(./images/3.1/bg.png) no-repeat;
            overflow: hidden;
            padding: 10px;
        }

        .king ul {
```

```

        overflow: hidden;
    }

    .king li {
        position: relative;
        float: left;
        width: 69px;
        height: 69px;
        margin-right: 10px;
    }

    .king li.current {
        /* width: 224px; */
    }

    .king li.current .big {
        /* display: block; */
    }

    .king li.current .small {
        /* display: none; */
    }

    .big {
        width: 224px;
        display: none;
    }

    .small {
        position: absolute;
        top: 0;
        left: 0;
        width: 69px;
        height: 69px;
        border-radius: 5px;
    }
</style>

</head>

<body>
    <script src="./js/jquery3.5.1.min.js"></script>

    <div class="king">
        <ul>
            <li>
                <a href="#">
                    
                    
                </a>
            </li>
            <li>

                <a href="#">

```

```

        
        
    </a>
</li>
<li>
    <a href="#">
        
        
    </a>
</li>
<li>
    <a href="#">
        
        
    </a>
</li>
<li>
    <a href="#">
        
        
    </a>
</li>
<li>
    <a href="#">
        
        
    </a>
</li>
<li>
    <a href="#">
        
        
    </a>
</li>
</ul>
</div>
</body>
</html>

```

正确的js代码:

// 需求: 鼠标移入 li, 改变标签的宽度, 切换图片的显示

```

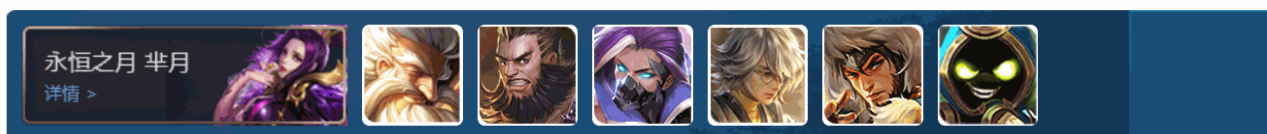
$("li").mouseover(function () {
    // 1.当前自己li 变宽
    $(this).stop().animate({width: "224px"});
    // 2.自己的小图-淡出
    $(this).find(".small").stop().fadeOut();
    // 3.大图看见-淡入
    $(this).find(".big").stop().fadeIn();
});

```

```
// 鼠标移出
$("li").mouseout(function () {
    // 1.当前自己li 变宽
    $(this).stop().animate({width: "69px"});
    // 小图-淡入
    $(this).find(".small").stop().fadeIn();
    // 大图-淡出
    $(this).find(".big").stop().fadeOut();
});
```

3.2 案例 - JQ版 - 查看英雄介绍2

与上标签和素材类似, 就是效果不同



正确的 js代码:

```
// 需求: 鼠标移入 li, 改变标签的宽度, 切换图片的显示
// 默认打开第一个是宽的(默认给第一个li, 设置class为current)
$("li:eq(0)").addClass("current");

$("li").mouseover(function () {
    // 1.当前自己li 变宽
    $(this).stop().animate({width: "224px"});
    // 2.自己的小图-淡出
    $(this).find(".small").stop().fadeOut();
    // 3.大图看见-淡入
    $(this).find(".big").stop().fadeIn();

    // 兄弟们li, 宽度变小, li下的大图淡出, 小图淡入
    $(this).siblings().stop().animate({width: "69px"});
    // 兄弟们都是大图淡出, 小图淡入
    $(this).siblings().find(".small").stop().fadeIn();
    $(this).siblings().find(".big").stop().fadeOut();
});
```

作业

作业1-使用jQuery完成手风琴效果

素材在images/下看文件夹名查找

效果演示:



标签和模板

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>手风琴</title>
  <style>
    ul {
      list-style: none;
    }

    * {
      margin: 0;
      padding: 0;
    }

    div {
      width: 1200px;
      height: 400px;
      margin: 50px auto;
      border: 1px solid red;
      overflow: hidden;
    }

    div li {
      width: 240px;
      height: 400px;
      float: left;
      /* transition: width 500ms; */
    }

    div ul {
      width: 1300px;
    }

    img {
      width: 800px;
```

```

    }
    </style>
</head>

<body>
    <div id="box">
        <ul>
            <li>
                <a href="#">
                    
                </a>
            </li>
            <li>
                <a href="#">
                    
                </a>
            </li>
            <li>
                <a href="#">
                    
                </a>
            </li>
            <li>
                <a href="#">
                    
                </a>
            </li>
            <li>
                <a href="#">
                    
                </a>
            </li>
        </ul>
    </div>
</body>
<script src="./js/jquery3.5.1.js"></script>
<script>
</script>

</html>

```

正确js代码

```

// 需求：鼠标移入时，当前li宽度放大，其余缩小，移出时，所有人恢复最初状态
// 思路：获取所有li，分别绑定移入事件，移入所有人先设置100宽度，当前li单独设置800宽，移出时，所有人恢复成最开始240px宽度

// 1. 每个li - 鼠标移入
$("#box li").mouseenter(function(){
    // 2. 让兄弟们变成100px
    $(this).siblings().stop().animate({"width": "100px"});

    // 3. 当前li - 宽度800px

```

```
$(this).stop().animate({"width": "800px"});
}).mouseleave(function(){
    // 4. 每个li - 鼠标移出
    $("#box li").stop().animate({"width": "240px"});
})

// // 1. 获取标签
// var lis = document.querySelectorAll('#box li');
// // 2. 产生索引
// for (var i = 0; i < lis.length; i++) {
//     // 2.1 每个li - 鼠标移入
//     lis[i].onmouseenter = function(ev){
//         // 2.2 把所有li设置100px宽
//         for (var j = 0; j < lis.length; j++) {
//             lis[j].style.width = "100px";
//         }
//         // 2.3 移入的这个li, 单独设置宽度800px宽
//         this.style.width = "800px";
//     }

//     // 2.4 每个li - 鼠标移出
//     lis[i].onmouseleave = function(ev){
//         // 2.5 所有li设置240px宽
//         for (var j = 0; j < lis.length; j++) {
//             lis[j].style.width = "240px";
//         }
//     }
// }
// }
```

作业2 - 完成商品切换查看效果

图片素材在images/看文件夹名找

效果演示:



标签和样式模板:

```
<!DOCTYPE html>
<html>

<head lang="en">
  <meta charset="UTF-8">
  <title></title>
  <style type="text/css">
    * {
      margin: 0;
      padding: 0;
      font-size: 12px;
    }

    ul {
      list-style: none;
    }

    a {
      text-decoration: none;
    }

    .wrapper {
      width: 250px;
      height: 248px;
      margin: 100px auto 0;
      border: 1px solid pink;
```



```

        border-right: 0;
        overflow: hidden;
    }

    #left,
    #content {
        float: left;
    }

    #left li {
        background: url(images/lili.jpg) repeat-x;
    }

    #left li a {
        display: block;
        width: 48px;
        height: 27px;
        border-bottom: 1px solid pink;
        line-height: 27px;
        text-align: center;
        color: black;
    }

    #left li a:hover {
        background-image: url(images/abg.gif);
    }

    #content {
        border-left: 1px solid pink;
        border-right: 1px solid pink;
    }
</style>
<script src="jquery.min.js"></script>
<script>

</script>
</head>

<body>
    <div class="wrapper">
        <ul id="left">
            <li><a href="#">女靴</a></li>
            <li><a href="#">雪地靴</a></li>
            <li><a href="#">冬裙</a></li>
            <li><a href="#">呢大衣</a></li>
            <li><a href="#">毛衣</a></li>
            <li><a href="#">棉服</a></li>
            <li><a href="#">女裤</a></li>
            <li><a href="#">羽绒服</a></li>
            <li><a href="#">牛仔裤</a></li>
        </ul>
        <div id="content">

            <div>

```

```

        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
    </div>

</div>

</div>
</body>

</html>

```

正确的js代码

```

$(function() {
    // 1. 鼠标经过左侧的小li
    $("#left li").mouseover(function() {
        // 2. 得到当前小li 的索引号
        var index = $(this).index();
        console.log(index);
        // 3. 让我们右侧的盒子相应索引号的图片显示出来就好了
        // $("#content div").eq(index).show();
        // 4. 让其余的图片（就是其他的兄弟）隐藏起来
        // $("#content div").eq(index).siblings().hide();
        // 链式编程
        $("#content div").eq(index).show().siblings().hide();
    })
})

```

Day02

课上练习

暂无

案例

实现整个购物车效果 - 标签和样式已经全部在这里了

在images/看序号, 找到模板项目 - 基于这个来实现下面的功能



品优购 购物车
pinyou.com

全部商品

<input type="checkbox"/> 全选	商品	单价	数量	小计	操作
<input type="checkbox"/>	 <div>【5本26.8元】经典儿童文学彩图青少年八十天环游地球中学生语文教学大纲</div>	¥12.60	<div>- 1 +</div>	¥12.60	删除
<input type="checkbox"/>	 <div>【2000张贴纸】贴纸书 3-6岁 贴画儿童 贴画书全套12册 贴画 贴纸儿童 汽</div>	¥24.80	<div>- 1 +</div>	¥24.80	删除
<input type="checkbox"/>	 <div>唐诗三百首+成语故事全2册 一年级课外书 精装注音儿童版 小学生二三年级课外阅读书籍</div>	¥29.80	<div>- 5 +</div>	¥149.00	删除

☐ 全选 删除选中的商品 清理购物车

已经选 0 件商品 总价: ¥0.00

去结算

5.1 案例 - 购物车1 - 全选

```
// 1. 全选影响小选框
// 1.1 给全选框绑定点击事件 (注意全选框2个, 都叫.checkall)
$(".checkall").click(function(){ // 因为让this指向全选框-不能用箭头函数
    // 1.2 获取全选的状态 - $(this).prop("checked")
    // 1.3 获取所有小选框, 都设置成大选框的状态
    $(".cart-item .j-checkbox").prop("checked", $(this).prop("checked"));
    // 1.4 把大选框的都影响到
    $(".checkall").prop("checked", $(this).prop("checked"));
})
```

5.2 案例 - 购物车2 - 小影响全

```
// 2. 小选框 -> 全选
// 2.1 获取所有小选框，绑定点击事件
$(".cart-item .j-checkbox").click(()=>{
    // 2.2 把小选框集合JQ对象，转成数组类型
    // 2.3 使用every() 判断是否有不符合return条件的
    var isAll = Array.from($(".cart-item .j-checkbox")).every(el => el.checked == true)
    // 2.4 筛选的条件影响全选框
    $(".checkall").prop("checked", isAll);
})
```

5.3 案例 - 购物车3 - 选中商品变背景色

```
// 1. 全选 -> 小选框
$(".checkall").click(function(){
    $(".cart-item .j-checkbox").prop("checked", $(this).prop("checked"));
    $(".checkall").prop("checked", $(this).prop("checked"));
    // 3.1 选中商品改背景色/取消背景色
    if ($(this).prop("checked")) {
        $(".cart-item").addClass("check-cart-item");
    } else {
        $(".cart-item").removeClass("check-cart-item");
    }
})

// 2. 小选框 -> 全选
$(".cart-item .j-checkbox").click(function(){
    var isAll = Array.from($(".cart-item .j-checkbox")).every(el => el.checked == true)
    $(".checkall").prop("checked", isAll);

    // 3.2 选中商品改变背景色 - 取消选中取消背景色
    if ($(this).prop("checked")) {
        $(this).parent().parent().addClass("check-cart-item");
    } else {
        $(this).parent().parent().removeClass("check-cart-item");
    }
})
```

5.5 案例 - 购物车4 - 修改商品数量

· 基础知识:

o `substring`

o `toFixed` :控制小数位数，若没有则补0;

```
// 4. 修改商品数量和小计
// 4.1 给所有的+绑定点击事件
$(".increment").click(function () {
    // 4.2 对应输入框+1
    $(this).prev().val($(this).prev().val() * 1 + 1);

    // 4.5 获取单价，计算小计
```

```

    var price = $(this).parents(".p-num").prev(".p-price").html().substring(1); // ¥21.34 截取单
    价
    $(this).parents(".p-num").next(".p-sum").html("¥" + (price *
$(this).prev().val()).toFixed(2));
})
// 4.3 给所有的-绑定点击事件
$(".decrement").click(function () {
    // 4.4 如果值大于1, 才能执行-1
    if ($(this).next().val() > 1) {
        $(this).next().val($(this).next().val() - 1);
        // 4.6 获取单价, 计算小计
        var price = $(this).parents(".p-num").prev(".p-price").html().substring(1); // ¥21.34 截
    取单价
        $(this).parents(".p-num").next(".p-sum").html("¥" + (price *
$(this).next().val()).toFixed(2));
    }
})

```

5.6 案例 - 购物车5 - 删除单个商品

```

// 5. 删除单个商品
$(".p-action").click(function(){
    // 5.1 找到商品容器删除
    $(this).parents(".cart-item").remove();
    // 5.2 更新全选状态
    var isAll = Array.from($(".cart-item .j-checkbox")).every(el => el.checked == true)
    $(".checkall").prop("checked", isAll);
})

```

5.7 案例 - 购物车6 - 删除选中商品和清空

```

// 6. 删除选中的商品
// 6.1 给删除选中商品标签绑定点击事件
$(".remove-batch").click(function(){
    // 6.2 获取所有商品容器, 判断哪个容器的小多选框是选中状态, 则删除对应商品容器
    $(".cart-item").each(function(i, contentDiv){
        if ($(contentDiv).find(".j-checkbox").prop("checked")) {
            $(this).remove();
        }
    })
    // 6.3 更新全选状态
    $(".checkall").prop("checked", $(".cart-item").length != 0);
})

// 6.4 清空购物车绑定点击事件
$(".clear-all").click(function(){
    $(".cart-item").remove();
    $(".checkall").prop("checked", $(".cart-item").length != 0);
})

```

5.8 案例 - 购物车7 - 已选总数量

```
// 7. 已选总数量
// 全选 / 小选点击 / 点击商品数量 / 单个删除 / 删除选中 / 清空购物车 - 都得影响选中鼠数量，所以封装个方法，每个地方都调用
function totalNum(){
    // 7.1 判断当前商品是否勾选，选中累计数量
    var sum = 0;
    $(".cart-item").each(function(i, contentDiv){
        if ($(contentDiv).find(".j-checkbox").prop("checked")) {
            sum += $(contentDiv).find(".itxt").val() * 1;
        }
    })
    // 7.2 赋予到相应位置上
    $(".amount-sum").find("em").html(sum);
}
```

5.9 案例 - 购物车8 - 累计总价

```
// 7. 已选总数量
// 全选 / 小选点击 / 点击商品数量 / 单个删除 / 删除选中 / 清空购物车 - 都得影响选中鼠数量，所以封装个方法，每个地方都调用
function totalNum(){
    // 7.1 判断当前商品是否勾选，选中累计数量
    var sum = 0;
    // 8. 计算总价
    var allPrice = 0;
    $(".cart-item").each(function(i, contentDiv){
        if ($(contentDiv).find(".j-checkbox").prop("checked")) {
            sum += $(contentDiv).find(".itxt").val() * 1;
            allPrice += $(contentDiv).find(".p-sum").html().substring(1) * 1;
        }
    })
    // 7.2 赋予到相应位置上
    $(".amount-sum").find("em").html(sum);
    // 8. 总价赋予到标签上
    $(".price-sum").find("em").html("¥" + allPrice.toFixed(2));
}
```

作业

课上案例来一遍

Day03

课上练习

暂无

案例

6.2 案例 - 返回顶部

效果演示:



标签模板可复制

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>案例_点击返回顶部</title>
  <style>
    html {
      height: 2000px;
    }

    div {
      position: fixed;
      right: 50px;
      bottom: 50px;
      width: 200px;
      height: 30px;
      background-color: #ccc;
      display: none;
    }
  </style>
</head>
<body>
  <div>
    <a href="#>返回顶部</a>
  </div>
</body>
</html>
```

```
</style>
</head>

<body>
  <div id="myDiv">返回顶部</div>
</body>
<script src="./js/jquery3.5.1.js"></script>
</html>
```

正确的js代码:

```
// 需求: 页面滚动到指定位置的时候, 右下角返回顶部的div显示
// 思路:
// 1. 滚动行为监听, 注册给document
// 2. 获取卷入高度值? 和某个值 (自己定) 300 进行比较!
// 3. 大于 设置的值, 返回按钮就可以出现;
// 4. 点击按钮, 返回到顶部(document的scrollTop为0)

// 1. document - 滚动事件
$(document).scroll(function () {
  // 2. 判断document的卷入值
  if ($(document).scrollTop() > 300) {
    // 3. 让右下角 - 出现
    $("#div").stop().show();
  } else {
    // 4. 否则让div淡出
    $("#div").stop().hide();
  }
});

// 4. 点击按钮, 返回到顶部
$("#div").click(function () {
  // 带动画的返回: 动画滚动: 兼容写法: 火狐和谷歌不能设置给document;
  $("body,html").stop().animate({
    scrollTop: 0
  });
});
```

6.3 案例 - 电梯导航

标签和模板请在images/素材中 - 下看序号

效果演示:



```
$(function () {
    // 需求:
    // 1. 网页滚动到今日推荐出现 - 左侧电梯导航, 滚动到某一个区域, 左侧对应电梯导航高亮
    // 2. 点击右侧电梯导航 - 网页滚动到对应模块
    // 3. 点击后bug解决, 网页刷新如果在下面 - 电梯也应该出现 - 导航也应该对应

    // 3.0 节流阀变量创建
    var flag = true;
    var toolTop = $(".recommend").offset().top;
    toggleTool();

    function toggleTool() { // 判断电梯导航是否出现
        if ($(document).scrollTop() >= toolTop) {
            $(".fixedtool").fadeIn();
        } else {
            $(".fixedtool").fadeOut();
        }
    };
}

// 1.0 给window - 绑定滚动事件
$(window).scroll(function () {
    // 1.1 判断电梯导航是否出现
    toggleTool();
    if (flag) { // 只要flag为true进入
        // 1.2 页面滚动到某个内容区域, 左侧电梯导航小li相应添加和删除current类名
        $(".floor .w").each(function (i, ele) {
            if ($(document).scrollTop() + 1 >= $(ele).offset().top) {
                $(".fixedtool li").eq(i).addClass("current").siblings().removeClass();
            }
        })
    }
})
}
```

```
});

// 2.0 点击电梯导航页面可以滚动到相应内容区域
$(".fixedtool li").click(function () {
    flag = false; // 点击把flag弄成false, 防止滚动的代码进入, 导致电梯导航li一直跳
    // 2.1 取出对应div的top值 - 要滚动到的位置
    var current = $(".floor .w").eq($(this).index()).offset().top;
    // 2.2 页面动画滚动效果
    $("body, html").stop().animate({
        scrollTop: current
    }, function () {
        flag = true;
    });
    // 2.3 点击之后, 让当前的小li 添加current 类名 , 姐妹移除current类名
    $(this).addClass("current").siblings().removeClass();
})
})
```

7.1 案例 - todolist-初始化列表

效果演示:



标签和样式模板:

```
<!DOCTYPE html>
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=no" />
  <title>ToDoList-最简单的待办事项列表</title>
  <link rel="stylesheet" href="css/index.css">
</head>

<body>
  <header>
    <section>
      <label for="title">ToDoList</label>

      <input type="text" id="title" name="title" placeholder="添加ToDo" required="required"
```

```

autocomplete="off" />
    </section>
</header>
<section>

    <h2>正在进行 <span id="todocount"></span></h2>
    <ol id="todolist" class="demo-box">
        <!-- <li><input type="checkbox">
            <p>任务名字</p>
            <a href="javascript:;"></a>
        </li> -->
    </ol>

    <h2>已经完成 <span id="donecount"></span></h2>
    <ul id="donelist">

    </ul>

</section>
<footer>
    Copyright &copy; 2014 todolist.cn
</footer>

<script src="./js/jquery.min.js"></script>

</body>

</html>

```

正确的js功能代码:

```

// 1.1 准备初始化任务数据
var arr = [{
    info: "学习",
    done: false
}, {
    info: "写代码",
    done: false
}, {
    info: "睡觉",
    done: true
}];

// 1.2 铺设初始任务
arr.forEach(function (obj, i) {
    var theLi = `${<li><input type="checkbox" ${obj.done? 'checked': ''}>
<p>${obj.info}</p>
<a href="javascript:;" index=${i}></a></li>`;
    // 已经完成

```

```

    if (obj.done) {
        $("#donelist").append(theLi);
    }
    // 没有完成
    else {
        $("#todolist").append(theLi);
    }
}
})

// 1.3 统计数量
$("#todocount").html($("#todolist li").length);
$("#donecount").html($("#donelist li").length);

```

7.2 案例 - todolist - 新增任务

· 和微博新增：思路相同

```

$("#title").on("keydown", function (ev) {
    if (ev.keyCode == 13 && $("#title").val().length > 0) { // 判断是回车
        var newObj = { // 新建数据对象
            info: $("#title").val(),
            done: false
        };
        dataArr.push(newObj); // 同步到数组中管理
        $("#title").val(""); // 清空输入框
        localStorage.setItem("todolist", JSON.stringify(dataArr));
        totalCount(); // 统计最新任务量
        load(); // 重新加载页面
    }
})

```

7.3 案例 - todolist - 删除任务

```

// 3.0 给a绑定点击事件 - 删除动作 - 事件委托
// 因为a是在上面动态创建的，所以事件委托既能给现在的a还能给将来的a委托一个点击事件
$("#todolist, #donelist").on("click", "a", function () {
    // 3.1 获取索引
    var index = $(this).attr("index");
    // 3.2 删除数组元素
    dataArr.splice(index, 1);
    // 3.3 更新到本地
    localStorage.setItem("todolist", JSON.stringify(dataArr));
    // 3.4 更新页面
    load();
})

```

7.4 案例 - todolist - 完成任务/取消完成任务

```
// 4.0 事件委托 - input - 点击事件
$("#todolist, #donelist").on("click", "input", function () {
    // 3.1 获取索引
    var index = $(this).next().next().attr("index");
    // 3.2 数据更新
    dataArr[index].done = !dataArr[index].done;
    // 3.3 更新到本地
    localStorage.setItem("todolist", JSON.stringify(dataArr));
    // 3.4 更新页面
    load();
})
```

7.5 案例 - todolist - 类的封装 - 优化

```
class TodoList {
    constructor() {
        if (localStorage['todolist']) { // 本地有数据
            this.dataArr = JSON.parse(localStorage['todolist']);
        } else { // 本地没数据
            this.dataArr = [];
        }
        this.load();
        $("#title").on("keydown", this.enterTodo.bind(this));
        $("#todolist, #donelist").on("click", "a", this.deleteTodo.bind(this));
        $("#todolist, #donelist").on("click", "input", this.changeTodo.bind(this));
    }
    load() {
        // 先清空页面现有的
        $("#todolist, #donelist").empty();
        // 开始循环现在数组
        this.dataArr.forEach(function (obj, i) {
            var theLi = `<li><input type="checkbox" ${obj.done ? 'checked' : ''}>
<p>${obj.info}</p>
<a href="javascript:;" index=${i}></a></li>`;
            // 已经完成
            if (obj.done) {
                $("#donelist").append(theLi);
            }
            // 没有完成
            else {
                $("#todolist").append(theLi);
            }
        })
        this.totalCount();
    }
    totalCount() {
        $("#todocount").html($("#todolist li").length);
        $("#donecount").html($("#donelist li").length);
    }
    enterTodo(ev) {
        if (ev.keyCode == 13 && $("#title").val().length > 0) { // 判断是回车
            var newObj = { // 新建数据对象
```

```

        info: $("#title").val(),
        done: false
    };
    this.dataArr.push(newObj); // 同步到数组中管理
    $("#title").val(""); // 清空输入框
    localStorage.setItem("todolist", JSON.stringify(this.dataArr));
    this.load(); // 重新加载页面
    }
}
deleteTodo(ev) {
    // 3.1 获取索引
    var index = $(ev.target).attr("index");
    // 3.2 删除数组元素
    this.dataArr.splice(index, 1);
    // 3.3 更新到本地
    localStorage.setItem("todolist", JSON.stringify(this.dataArr));
    // 3.4 更新页面
    this.load();
}
changeTodo(ev) {
    // 3.1 获取索引
    var index = $(ev.target).next().next().attr("index");
    // 3.2 数据更新
    this.dataArr[index].done = !this.dataArr[index].done;
    // 3.3 更新到本地
    localStorage.setItem("todolist", JSON.stringify(this.dataArr));
    // 3.4 更新页面
    this.load();
}
}
new TodoList();

```

8.0 案例 - JQ - 插件 - 瀑布流

· <http://www.htmleaf.com/jQuery/pubuliuchajian/201601093003.html>

- 去网站找适合demo 插件，看演示！
- 场景：个人博客图片展示啊！文章列表！
- 下载插件！插件是在JQ基础继续写的一些功能！
- o 插件**JS文件** 必须引入在**JQ文件后面！**
- 按照官方的步骤：
 - o 复制html、复制CSS；
 - o 初始化JS某个函数；

```

$("#gallery-wrapper").pinterest_grid(); // 参数配置都是默认；

// no_columns: 网格布局一行的列数。默认值为一行3个网格。
// padding_x: 网格在x轴方向的padding值。默认值为10像素。
// padding_y: 网格在y轴方向的padding值。默认值为10像素。

```

```
// margin_bottom: 网格底部的margin值。默认值为50像素。  
// single_column_breakpoint: 指定在视口多大时一行只显示一个网格。  
  
// 例如:  
$("#gallery-wrapper").pinterest_grid({  
  no_columns: 4,  
  padding_x: 15,  
  padding_y: 10,  
  margin_bottom: 50,  
  single_column_breakpoint: 700  
});
```

作业

暂无